Improving Scalbility in Metadata and Small File Accesses

Walt Ligon

walt@clemson.edu

CLEMSON UNIVERSITY

Areas of Investigation

- Problem focus
  - Metadata Operations (Create, Remove, Stat)
  - Small Files
  - Unaligned Accesses
  - Structured I/O
- Approaches
  - Server-to-server communication
  - Collective metadata ops
  - Middleware based caching

Research Thrusts

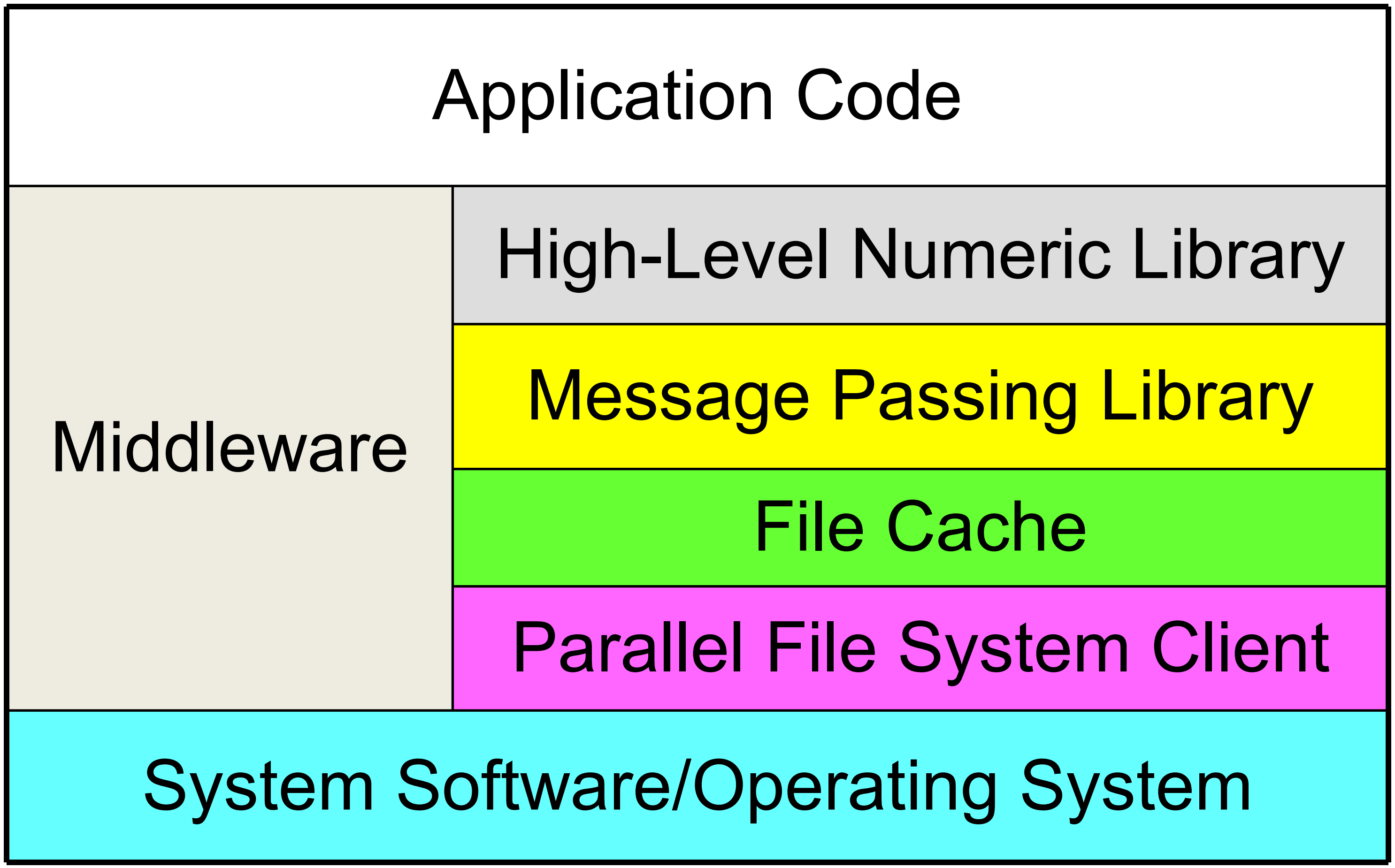
- Brad Settlemyer (PhD '09)
  - HECIOS simulator
  - Validation experiments
  - Study of middleware-based cache
- Yang Wu (MS '09)
  - Study of distributed directories
- Additional Projects

CLEMSON UNIVERSITY

Middleware Caching

- Improves locality
  - PVFS Acache and Ncache
  - Improve write-read and read-read accesses
- Small accesses
  - Can bundle small accesses into a single large operation
- Alignment
  - Can compress accesses by performing aligned requests
- Transparent to application programmer

Proposed Parallel Software Stack

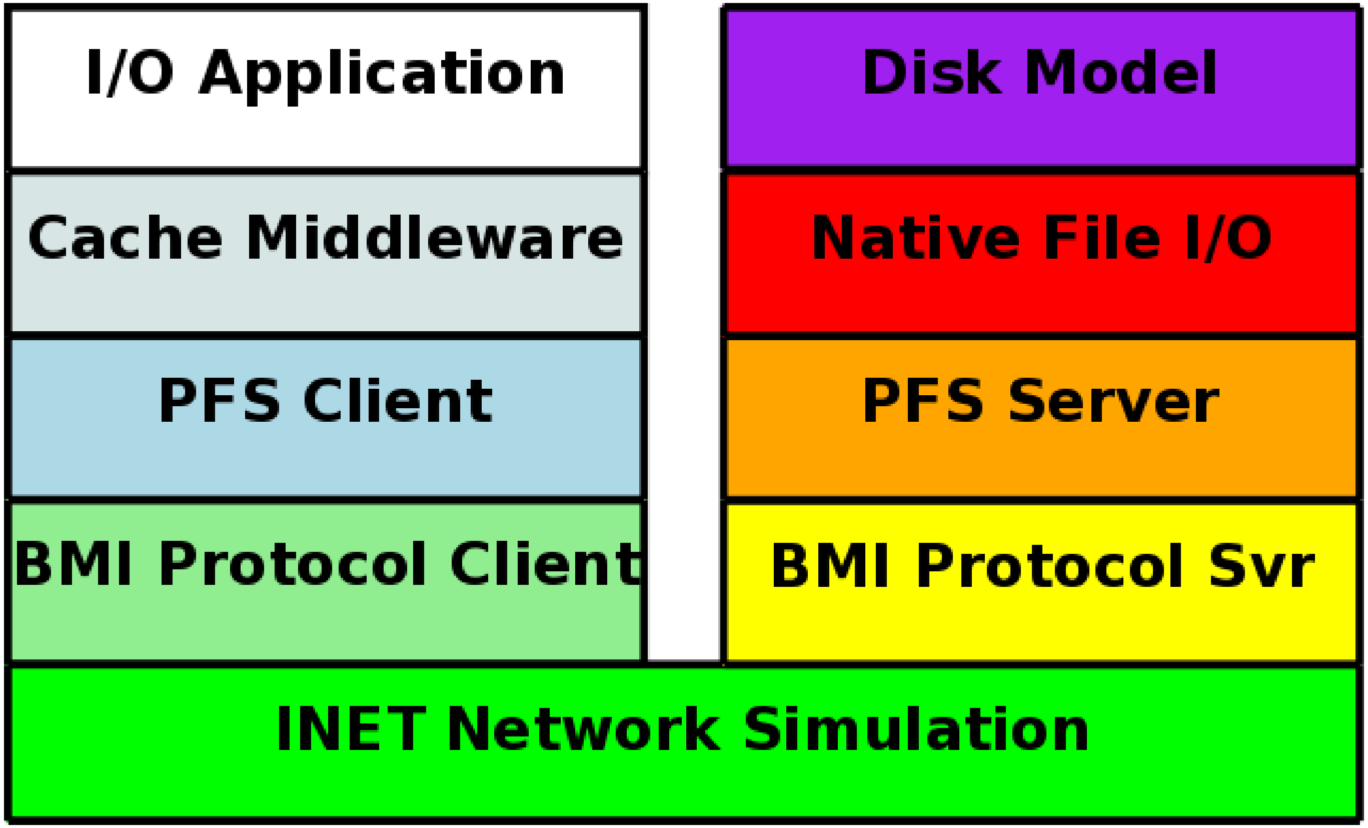


Middleware Caching Study

- Tuning experiments
- Benefit of shared multi-core cache?
- Benefit of different cache organizations?
  - Fixed-page data cache
  - Progressive page cache
  - File view-aware aggregation
- Simulation-based study: HECIOS
- Argonne MPI I/O Test
  - Bandwidth test reading/writing 16MB common file
- Flash I/O benchmark
  - Many small, unaligned file writes

CLEMSON UNIVERSITY

HECIOS Overview

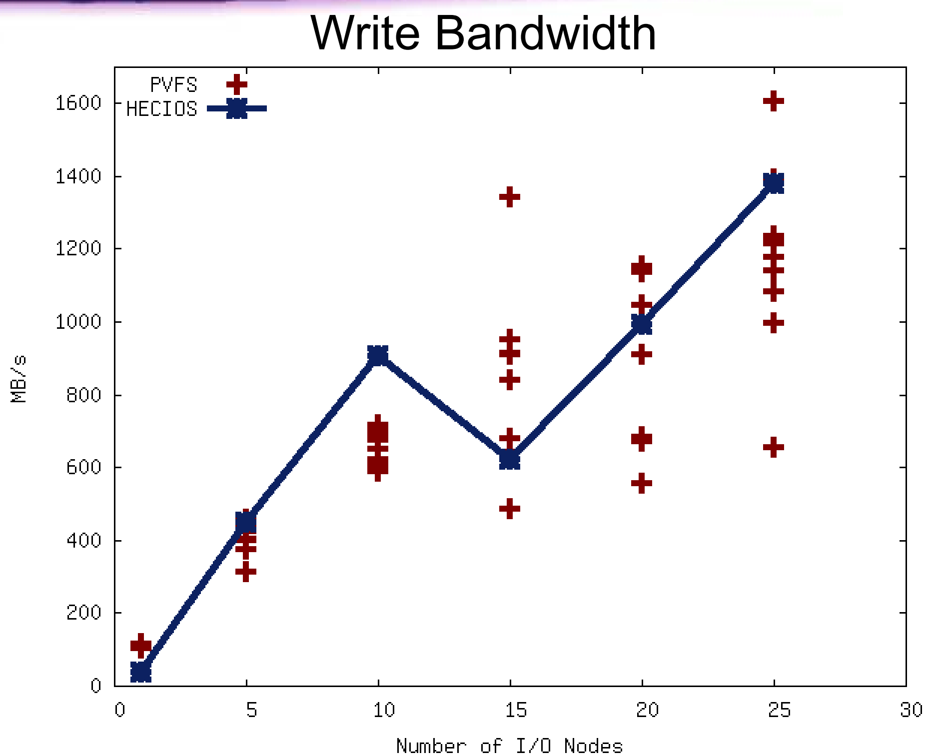


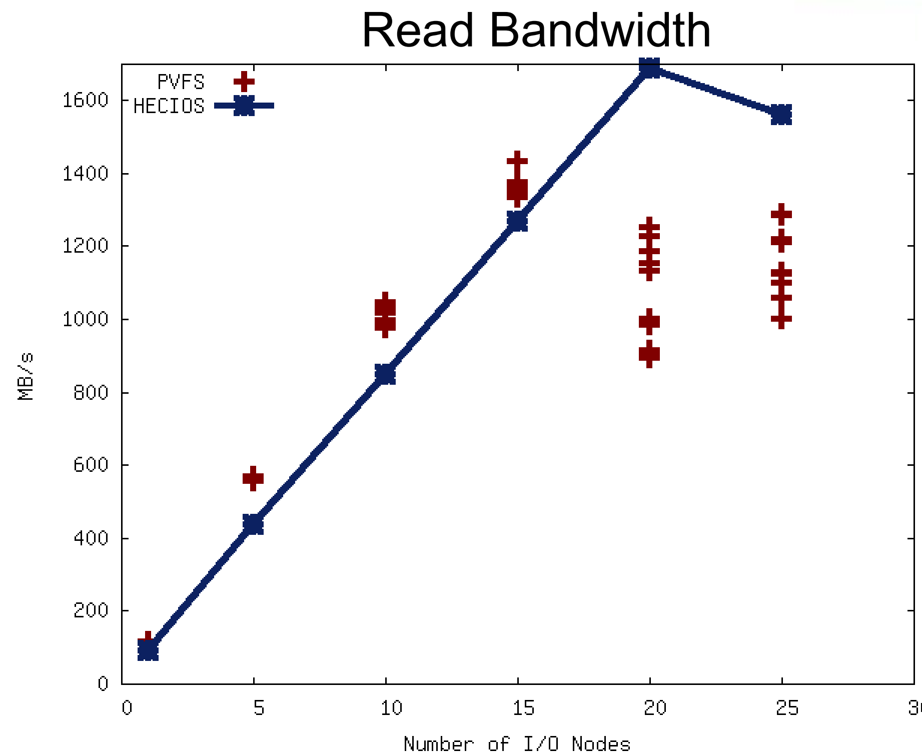
HECIOS System Architecture

Validation/Tuning Experiments

- Simulator designed to track PVFS performance
- Validation Experiments
  - Ping latency
  - File system bandwidth
  - Flash I/O (a small write benchmark)
  - Case study with metadata operations
- Myrinet/switch modeling discrepancies

Validation Bandwidth Results

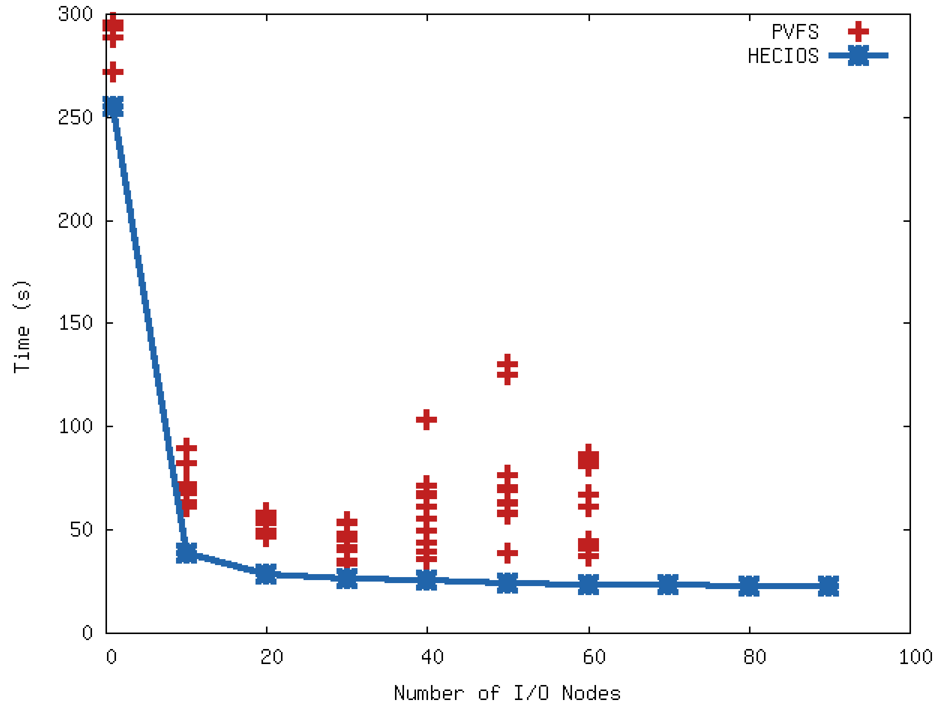




- Argonne I/O Benchmark (higher is better)
- 32 CPUN, 8PPN, Gigabit Ethernet

CLEMSON
UNIVERSITY

Validation Results

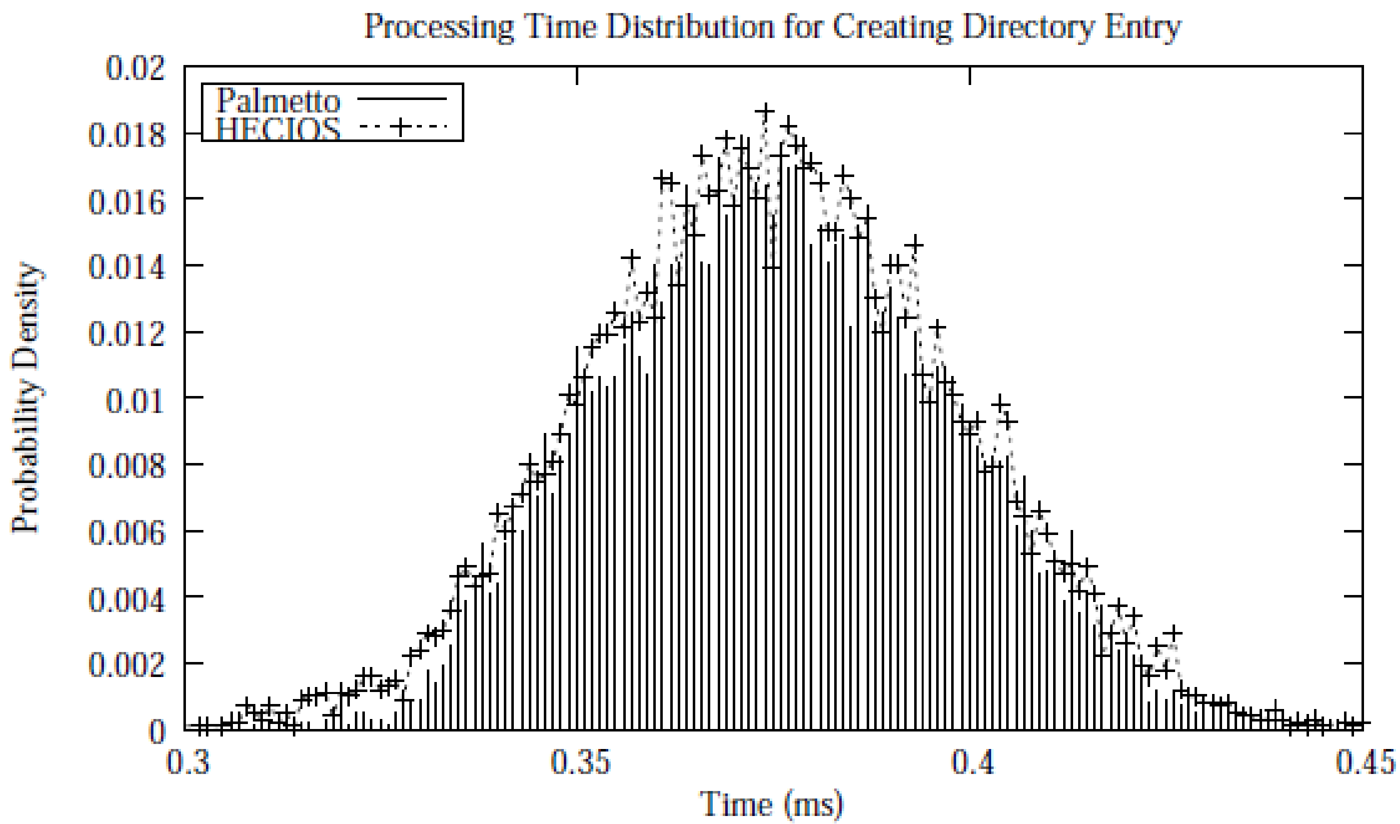


- FLASH I/O Benchmark (lower is better)
- 64 CPUN, 8PPN, Gigabit Ethernet

CLEMSON UNIVERSITY

Implementation – Modeling (cont.)

- Create Directory Entry Processing Time

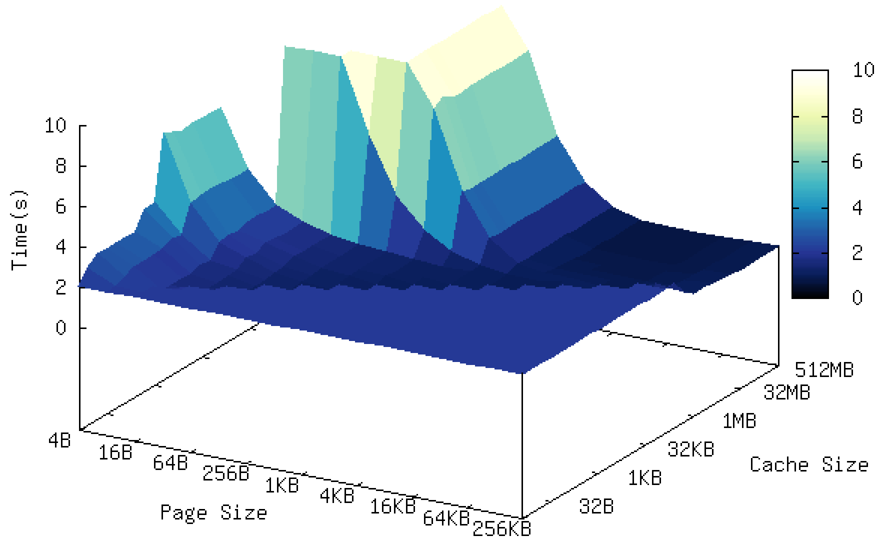


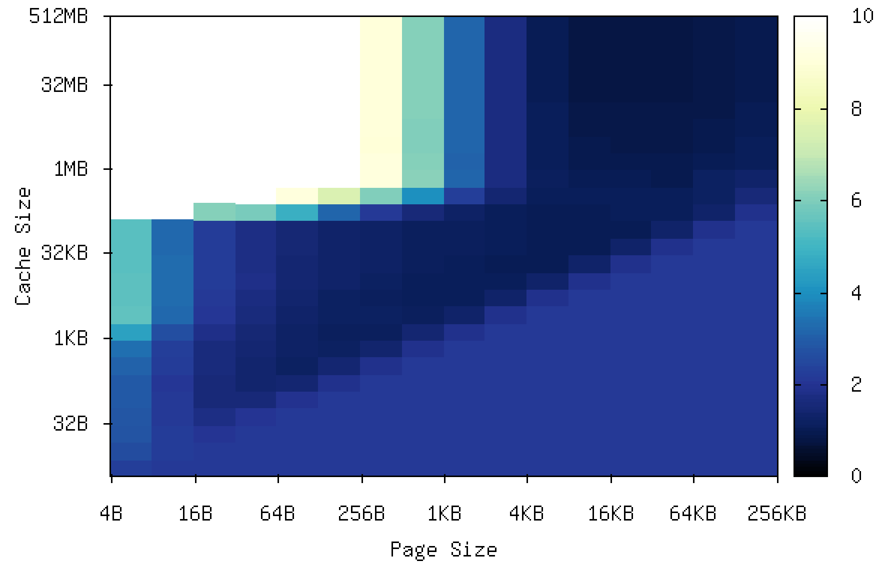
CLEMSON UNIVERSITY

Fixed-size Page Caching

- Benefits of paged caching
  - Extracts fixed-size update structure from codes that do not write data with structure
  - Efficient for the file system
  - Low cache metadata overhead
- Issues with paged caching
  - Poorly aligned page sizes may reduce performance
  - Must read page for partial page writes

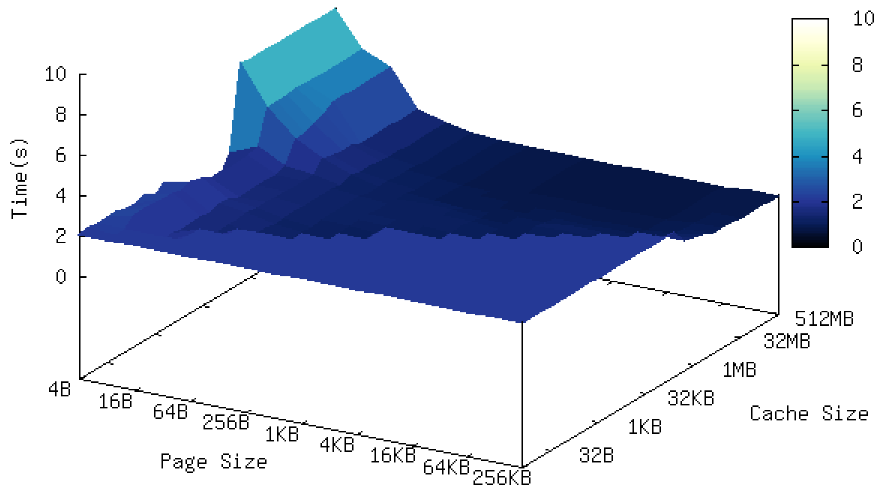
Baseline Cache Performance

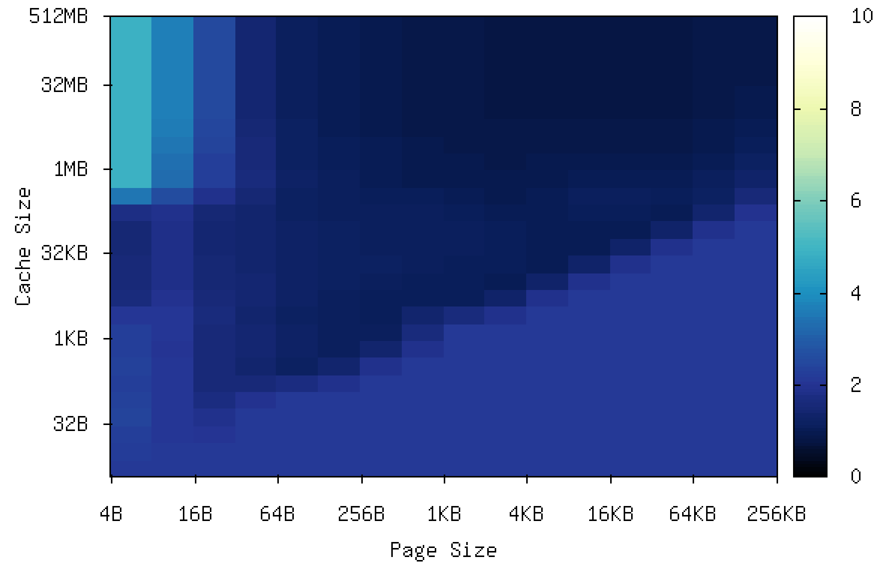




- Flash I/O Benchmark
- 8 CPUN, 1PPN, 4ION, Gigabit Ethernet
- Includes large write bypassing

Leveraging Request Data Types

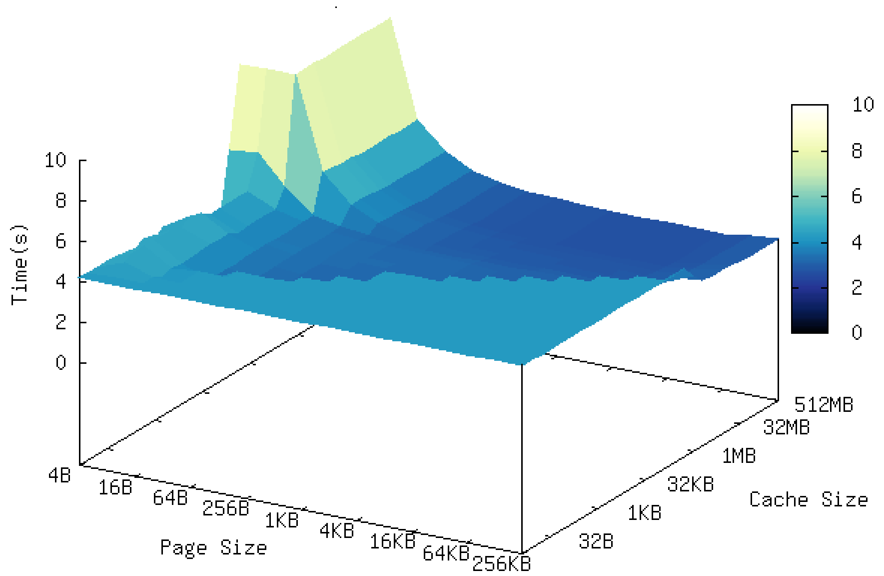


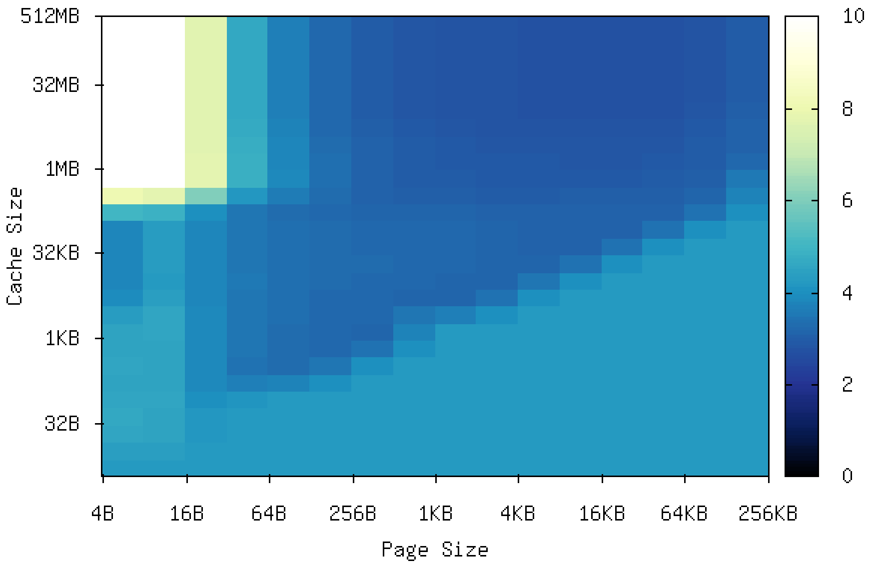


- Flash I/O Benchmark
- 8 CPUN, 1PPN, 4ION, Gigabit Ethernet
- Includes large write bypassing

CLEMSON
UNIVERSITY

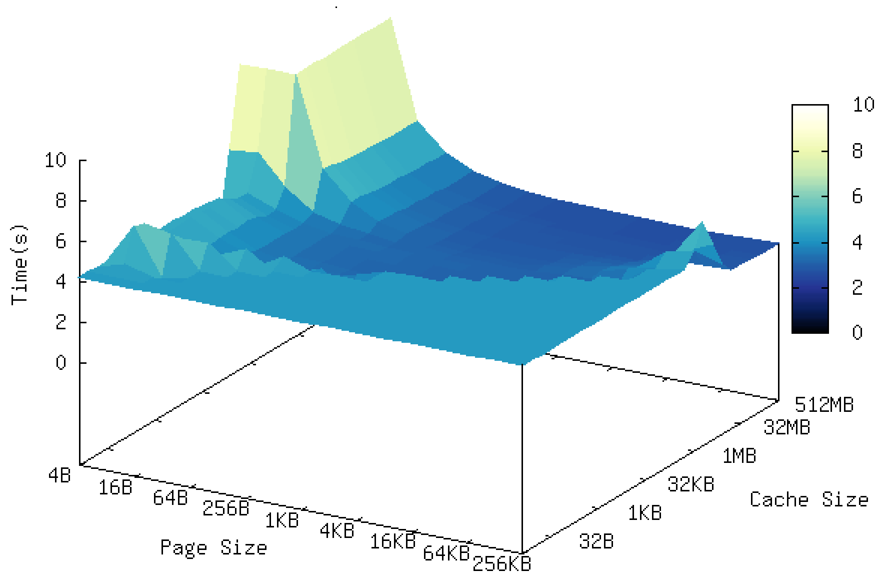
Multi-core Cache Performance

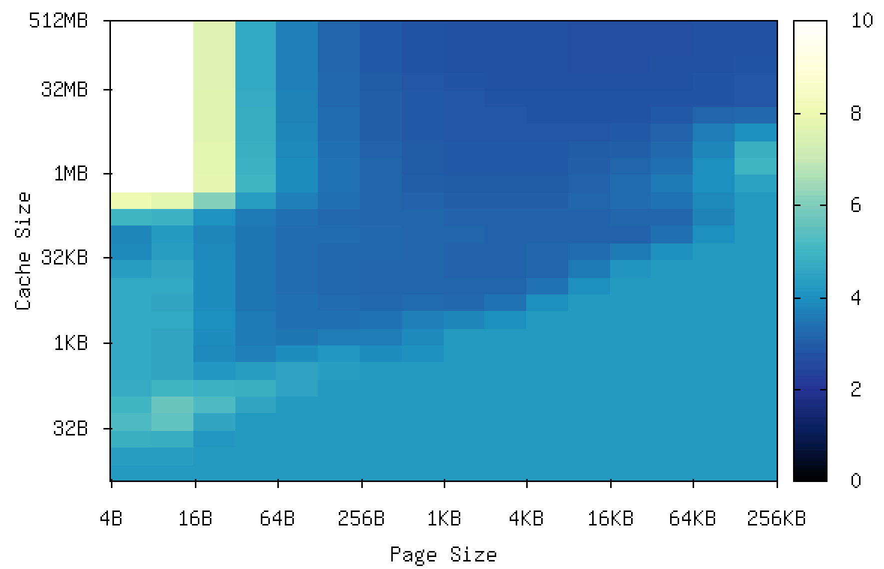




- Flash I/O Benchmark
- 1 CPUN, 8 PPN, 4 ION, Gigabit Ethernet
- Includes large write bypassing

Shared Cache Performance

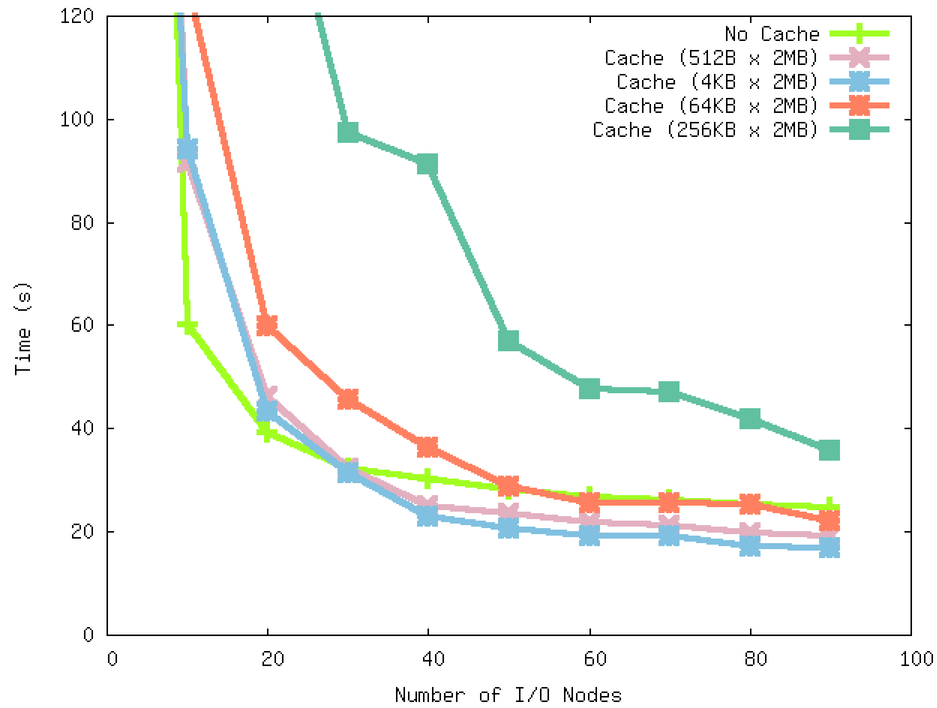


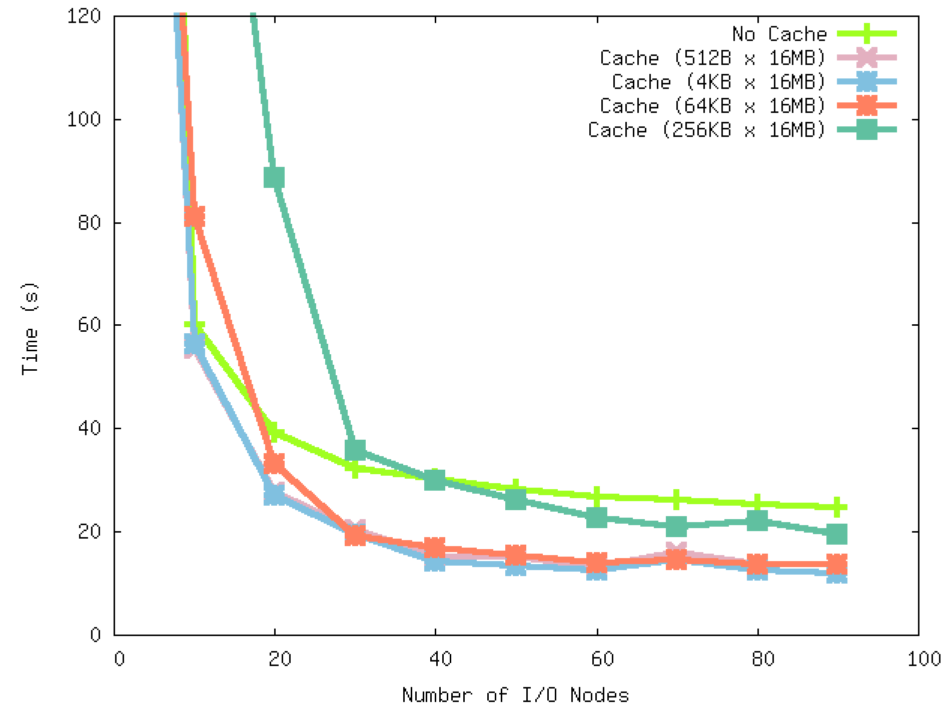


- Flash I/O Benchmark
- 1 CPUN, 8PPN, 4ION, Gigabit Ethernet
- Includes large write bypassing

CLEMSON UNIVERSITY

Large Scale Cache Performance





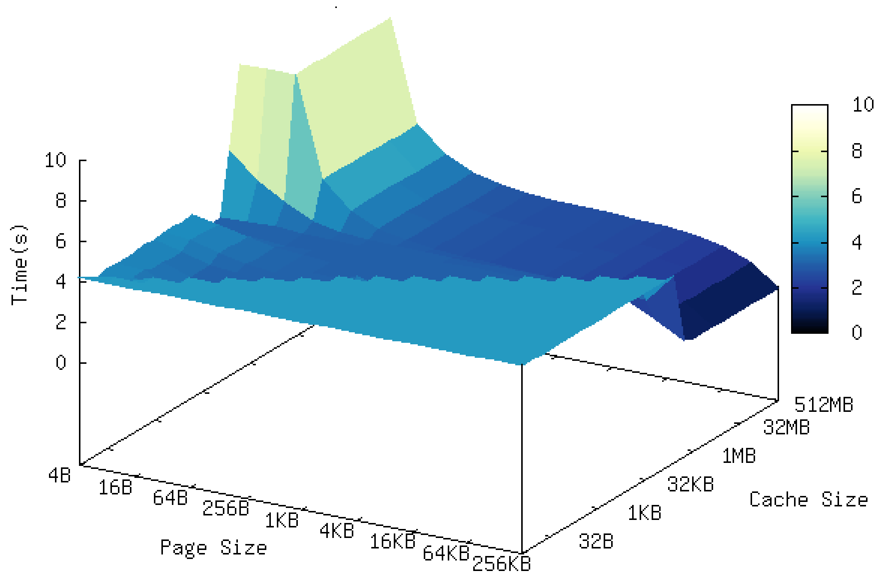
- Flash I/O Benchmark
- 128 CPUN, 8PPN, Gigabit Ethernet
- Shared fixed-size page cache

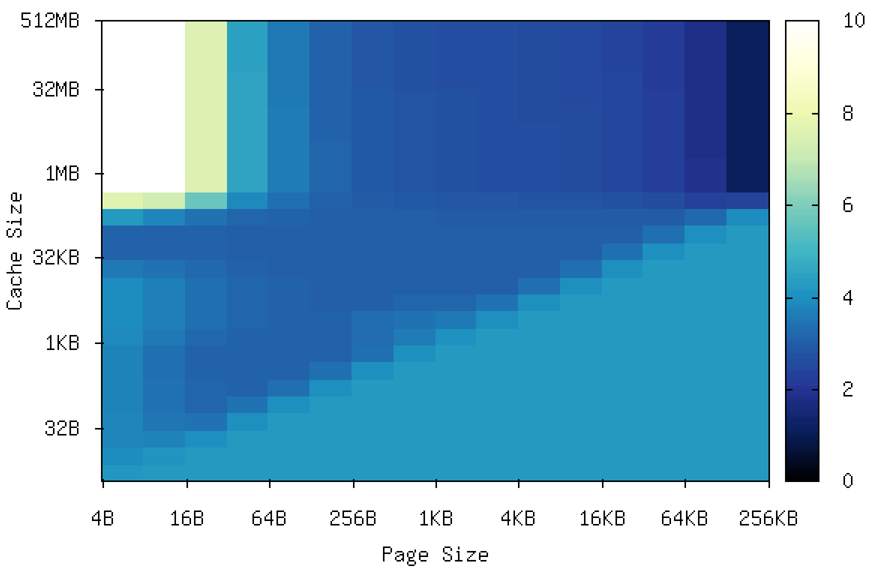
CLEMSON UNIVERSITY

Progressive Page Caching

- Eliminates performance penalties associated with false sharing avoidance
- Simpler cache concept
- Significant cache metadata management overhead
  - Dirty mask – fixed overhead
  - File Region Tree – dynamic overhead

Baseline Cache Performance

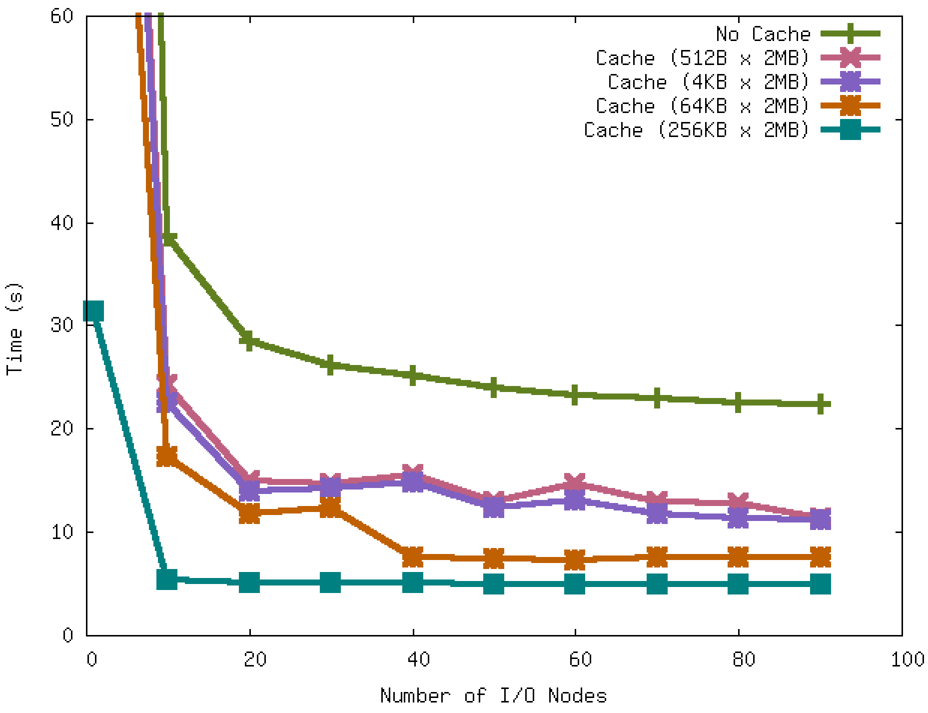


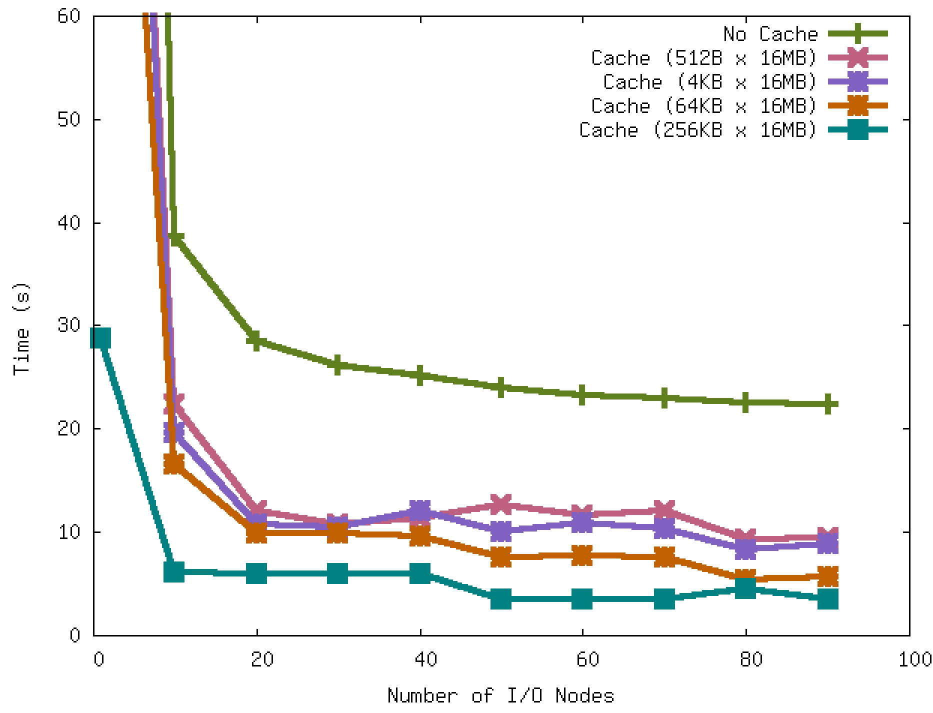


- Flash I/O Benchmark
- 8 CPUN, 1PPN, 4ION
- Includes large write bypassing

CLEMSON UNIVERSITY

Large Scale Cache Performance





- Flash I/O Benchmark
- 64 CPUN, 8PPN, Gigabit Ethernet
- Shared progressive page cache

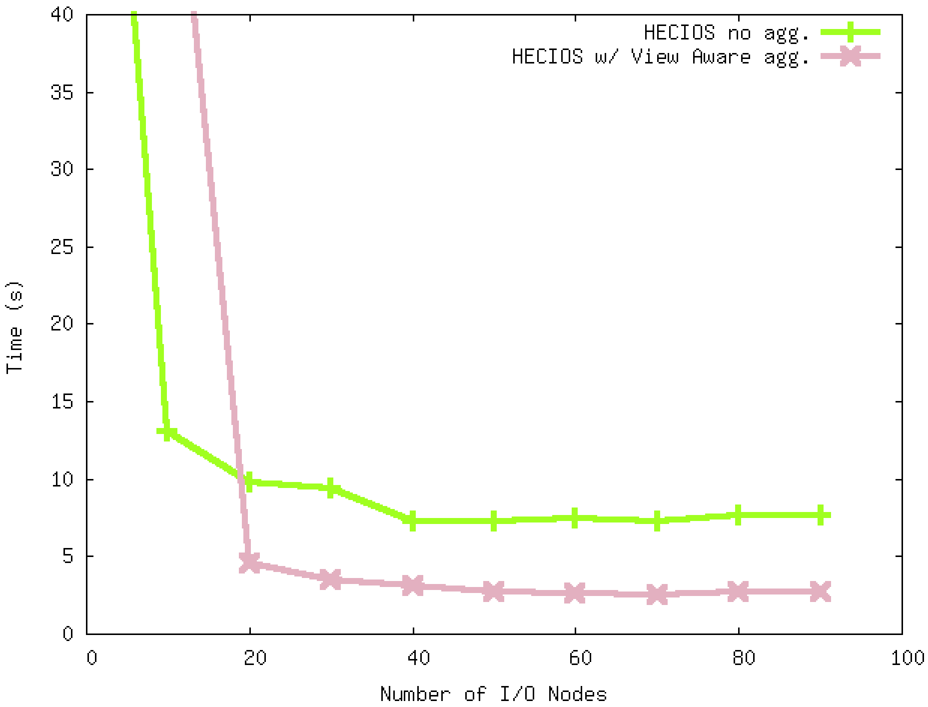
CLEMSON UNIVERSITY

File View Aware Aggregation

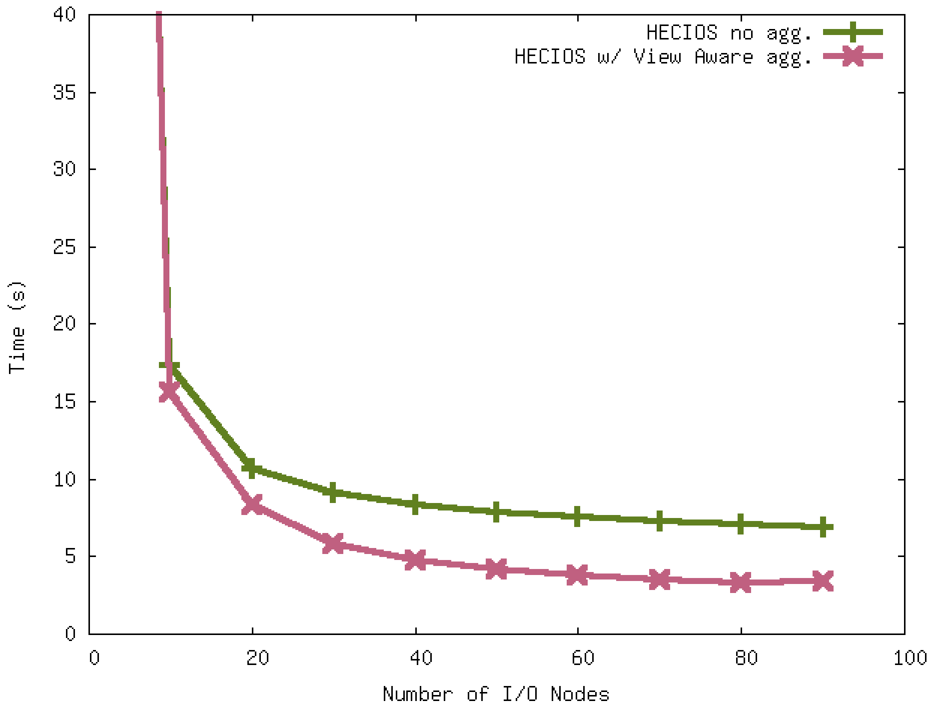
- MPI provides a more descriptive facility for describing file I/O
  - Collective I/O
  - MPI provides file views for describing file sub-regions
- Use file views as a mechanism for coalescing reads and writes during collective I/O
- How to take the union of multiple views.
  - Heuristic approach to detect cyclic I/O patterns
  - Allow user to provide own union code

View Aware Aggregation Results

Write Bandwidth



Read Bandwidth



- MPI Tile I/O Benchmark (lower is better)
- 64 CPUN, 8PPN, Gigabit Ethernet

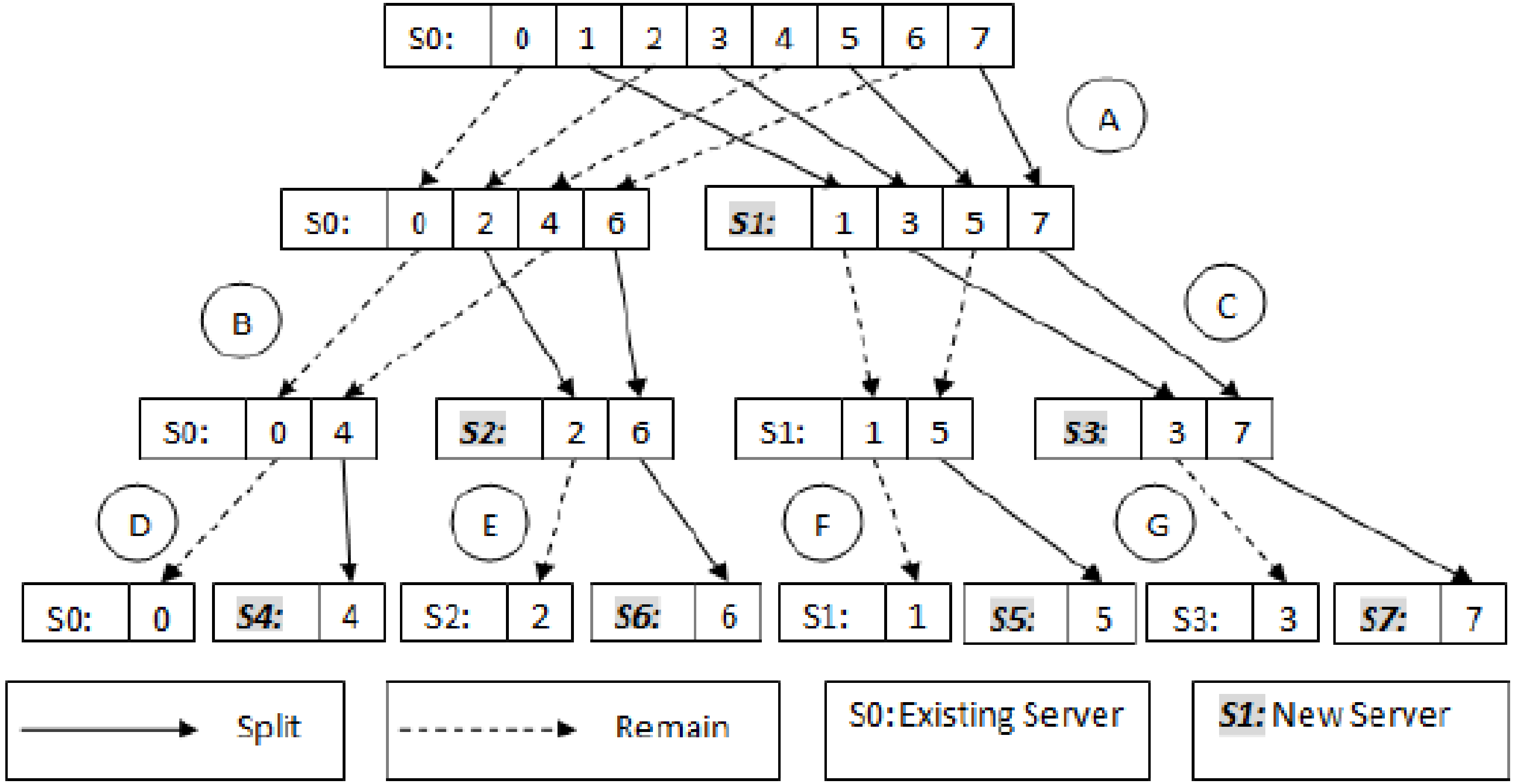
CLEMSON UNIVERSITY

Distributed Directory Study

- Distribute directory entries across servers
- P2S Map & Extensible Hashing
- Client Lookup
- Directory Splitting
- Request Forward & Response
- Directory Traversal

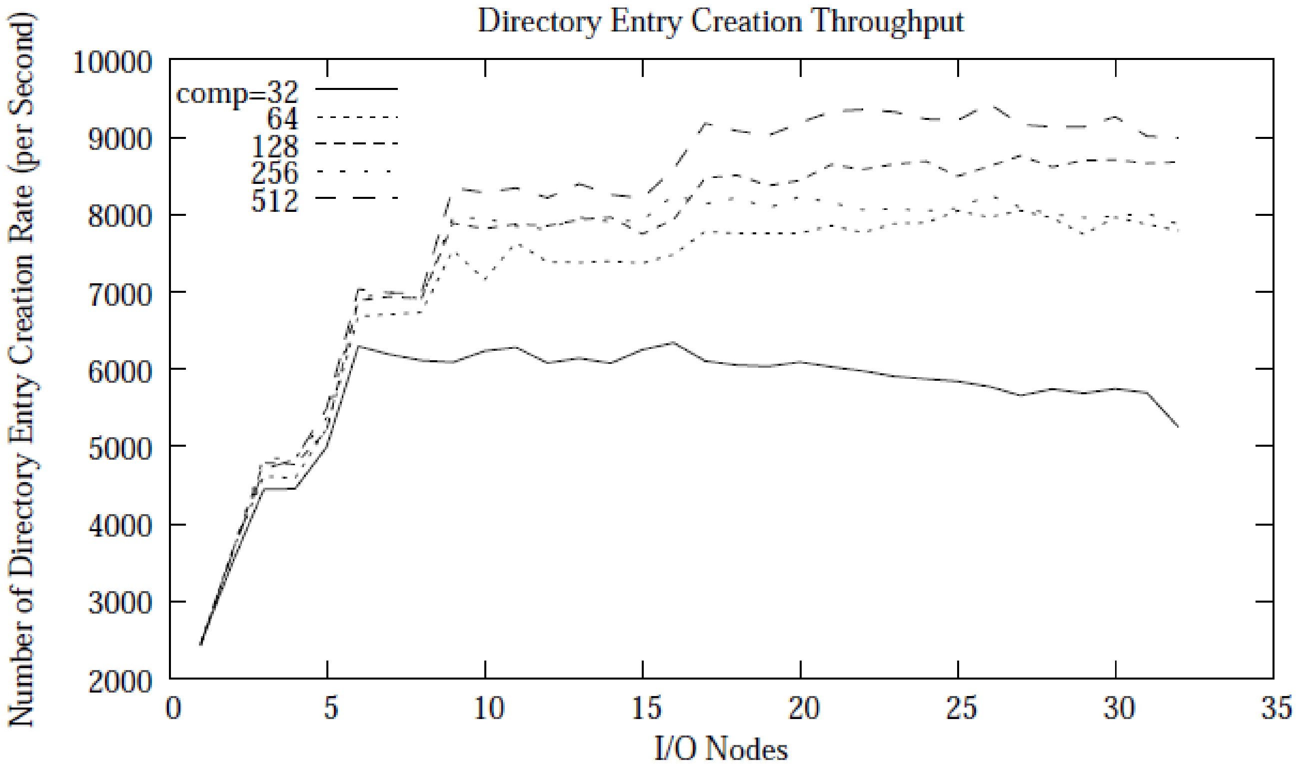
Extensible hashing

- Example:



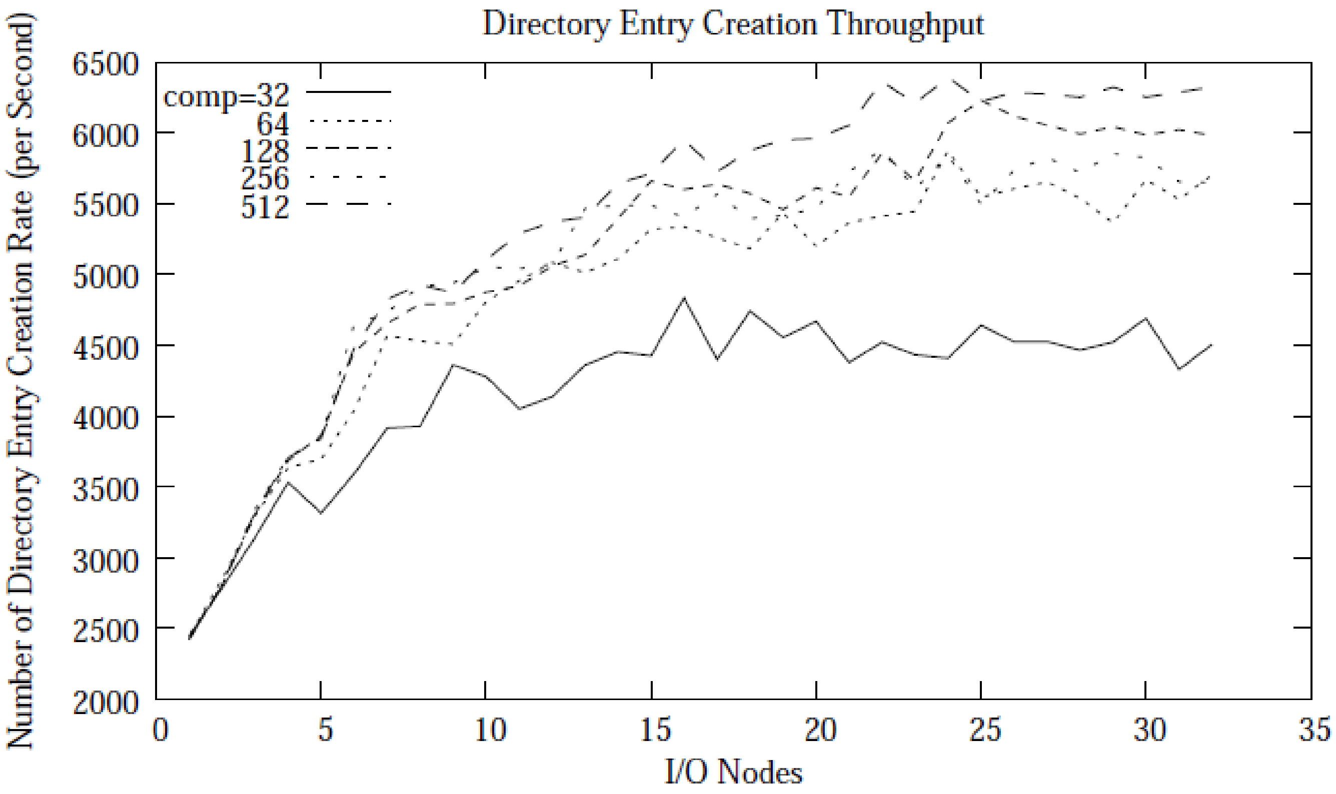
Experimental Results

- Throughput – Pre-distribution



Experimental Results

- Throughput – Dynamically Splitting



CLEMSON UNIVERSITY

Additional Development Projects

- Tree-based metadata operations
- Mirror-on-immutable
- Security prototype
  - Capability based
  - Timeouts/Revocations
  - Interface with various authentication sources
- SSD for metadata
- Lookup by attribute value

Projects still to come

- More simulations with distributed directory
- Prototype implementation of middleware cache
- Settlemyer at ORNL
  – larger simulations
- Would like ...
  – more disk models
  – more network models